Digital Scholarship in the Humanities Advance Access published December 15, 2015

ProppLearner: Deeply Annotating a Corpus of Russian Folktales to Enable the Machine Learning of a Russian Formalist Theory

Mark A. Finlayson
Florida International University, Miami, FL, USA

Abstract

I describe the collection and deep annotation of the semantics of a corpus of Russian folktales. This corpus, which I call the ‘ProppLearner’ corpus, was assembled to provide data for an algorithm designed to learn Vladimir Propp’s morphology of Russian hero tales. The corpus is the most deeply annotated narrative corpus available at this time. The algorithm and learning results are described elsewhere; here, I provide detail on the layers of annotation and how they were chosen, novel layers of annotation required for successful learning, the selection of the texts for annotation, the annotation process itself, and the resulting inter-annotator agreement measures. In particular, the corpus comprised fifteen texts totaling 18,862 words. There were eighteen layers of annotation, five of which were developed specifically to support learning Propp’s morphology: referent attributes, context relationships, event valences, Propp’s ‘dramatis personae’, and Propp’s functions. All annotations were created by trained annotators with the Story Workbench annotation tool, following a double-annotation paradigm. I discuss lessons learned from this effort and what they mean for future digital humanities efforts when working with the semantics of natural language text.

Correspondence: Mark A. Finlayson, School of Computing and Information Sciences, Florida International University, 11200 S.W. 8th Street, ECS Room 362, Miami, FL 33199, USA.
E-mail: markaf@fiu.edu

Successfully addressing humanist questions with computational techniques requires formalization at several different stages. The stage that is most often discussed is the last, where a computational learning algorithm or computer-implemented statistical technique is applied to the (humanist) data, producing results that bear on the question at hand. Nevertheless, as many digital humanists know, preparation of the data—namely, casting the data into a form suitable for computational analysis—is often the lion’s share of the work, and brings with it numerous theoretical assumptions and implicit biases. This data preparation is also often lightly treated, leaving large gaps in our understanding of the work described. Here, I attempt to address this deficiency for my own work and describe one such data formalization project, the construction of a corpus to support machine learning of a Russian formalist theory of narrative structure. The learning target was Vladimir Propp’s theory of folktale structure, his so-called *Morphology of the Folktale* (Propp, 1968), and I call the corpus the ‘ProppLearner’ corpus. I have described the overall success of the learning stage elsewhere (Finlayson, 2015), but

doi:10.1093/llc/fqv067

I gave only a thumbnail sketch of the data preparation stage of the project and the contents of corpus. Here, I reverse the emphasis: I endeavor to give a full view of the details of the data preparation but only provide a sketch of the learning results (§1.1). In the current article, I include key details on the layers of annotation and why they were chosen as well as novel layers of annotation required for successful learning; I further discuss in detail how the texts were selected, the annotation process itself, and the resulting inter-annotator agreement measures. I conclude the article with a discussion of the lessons learned from this effort, and how these lessons point the way forward for future digital humanities studies seeking to investigate the deep semantics of texts.

1 Learning Propp's Morphology

The actual problem I tackled was learning Propp's Morphology of the Folktale from text. Propp's theory is, at heart, what we now would describe as a 'plot grammar', which describes the types and order of abstracted plot pieces (which Propp called 'functions') that may occur in the folktales in his corpus. Propp's theory additionally described a gross level of tale organization (the 'move' structure), a set of long-distance dependencies between plot pieces (function subtypes), a number of exceptions and additional complications (order inversions and trebling), and common character types (the 'dramatis personae'). While Propp's theory was the first example of its kind, and was applied to a specific set of Russian folktales, later work showed how Propp's approach could be generalized and applied to different cultures (Colby, 1973; Dundes, 1964). As I describe elsewhere (Finlayson, 2015), being able to learn a morphology from data would be of great interest to many scholars and scientists, including folklorists, literary theories, cultural anthropologists, cultural psychologists, cognitive scientists, computational linguists, and researchers in artificial intelligence and machine learning.

The purpose of the work, therefore, was to learn Propp's theory from the folktales themselves, much in the way Propp himself did—to automate Propp's thinking. What, then, were the raw data of Propp's analysis? What did he use to generate his complex description of the underlying regularities of Russian hero tale plots? Propp read the folktales, of course, but it is clear that Propp's 'data' were more than just word counts or presence of common words and phrases. He was not looking at what might be called 'motifs' by folklorists or perhaps 'keywords' by computational linguists: he did not just look for all tales involving, say, 'sorcerers' and group them together, leaving them distinct from all tales involving 'tzars'; neither did he group together tales mentioning 'old' men, leaving tales involving 'young' men to another category. In fact, Propp specifically eschewed such an approach as one of the main sins of previous folktale indices, and emphasized that it leads to unprincipled and often uninformative categorization schemes. Instead, he insisted that we seek categories based on higher, more abstract groupings. He said:

> *Let us compare the following events*:
> *1. A tsar gives an eagle to a hero. The eagle carries the hero away to another kingdom.*
> *2. An old man gives Súčenko a horse. The horse carries Súčenko away to another kingdom.*
> *3. A sorcerer gives Ivan a little boat. The boat takes Ivan to another kingdom. [...]*
> *Both constants and variables are present in the preceding instances. The names of the* dramatis personae *change (as well as the attributes of each), but neither their actions nor functions change. From this we can draw the inference that a tale often attributes identical actions to various personages.... We shall have to determine to what extent these functions actually represent recurrent constants of the tale. (Propp, 1968, p. 20)*

Here, Propp's 'raw data', the data in which he is attempting to find patterns, are not motifs or keywords, but rather actors and actions. He is examining what I will call the 'surface semantics' of the tale, because it is the sort of meaning that is not deeply buried—it is the 'who does what to whom' of the tale. It is the sort of information that an elementary school student might be expected to take away from a reading. From these raw data, Propp sought to find commonalities and deeper patterns (the 'deep semantics', in contrast with surface semantics). My goal, therefore, when constructing the data to support learning Propp's morphology automatically, was to generate these 'raw' data, reflecting the surface semantics of the stories.

1.1 Learning results

Although my focus in this article is on the preparation of the data, I will digress for just a moment to reveal the result of the entire study, so that the reader is not left wondering to what end we are spending so much time and energy preparing our data.

The algorithm I developed is called 'Analogical Story Merging (ASM)', and is a modification of a grammar learning technique called model merging (Stolcke and Omohundro, 1994). In short, ASM starts with the most specific possible grammar for the data, and then uses similarities between portions of the data to generalize small parts of the grammar at a time. ASM maintains a score which is derived from the grammar's fit to the data, and finishes when it can no longer find generalizations which improve the score. ASM finds similarities between tales using a set of rules derived from Propp's descriptions of his own thought process.

Overall, the algorithm could be considered a qualified success. I used three different measures to analyze the performance of the algorithm. The first was the chance-adjusted Rand Index, a measure of the overall quality of the clustering of events into Propp's functions (Rota, 1964). On this measure, running from 1 (perfect match to Propp's results) to 0 (no correlation with Propp) to −1 (complete opposite of Propp), the algorithm achieves a score of between 0.511 and 0.714, depending on how it is calculated. The second measure was individual F_1-measures for each of Propp's functions. The most notable successes on this measure were the identification of Propp's functions of Struggle and Victory ('H & I'), Villainy/Lack ('A') and Reward ('W'), with F_1-measures all above 0.8. Other functions were retrieved with qualifiers proportional to the number of instances found in the data. The final metric was a cross-validation analysis of how well the implementation works with smaller amounts of data, which shows that the algorithm's performance over smaller amounts of data remains relatively robust.

2 Composition of the Data

What information, then, comprises the data—i.e. the surface semantics of the tales? Here, we have come to the heart of the data preparation problem. What information is included will determine, whether the endeavor is successful (learning Propp's morphology), how much of it can be learned, and what biases might be present in the final result. It also explicitly lays out my theoretical position, potentially leaving me open to charges of inaccuracy or infidelity (to Propp's purposes) in later critiques. It is critical that one chooses this information carefully.

What information do we need to represent explicitly to automate learning Propp's morphology? Propp himself discusses this early in his monograph:

> *...the functions of the dramatis personae are basic components of the tale, and we must first of all extract them. In order to extract the functions we must define them. Definition must proceed from two points of view. First of all, definition should in no case depend on the personage who carries out the function. Definition of a function will most often be given in the form of a noun expressing an action (interdiction, interrogation, flight, etc.). Secondly, an action cannot be defined apart from its place in the course of narration. The meaning which a given function has in the course of action must be considered. For example, if Ivan marries a tsar's daughter, this is something entirely different than the marriage of a father to a widow with two daughters. A second example: if, in one instance, a hero receives money from his father in the form of 100 rubles and subsequently buys a wise cat with this money, whereas in a second case, the hero is rewarded with a sum of money for an accomplished act of bravery (at which point the tale ends), we have before us two morphologically different elements—in spite of the identical action (the transference of money) in*

> *both cases. Thus, identical acts can have different meanings, and vice versa. Function is understood as an act of a character, defined from the point of view of its significance for the course of the action.* (Propp, 1968, p. 21)

In the following subsections, I outline in detail the layers of information that were explicitly captured to reflect Propp's attention. First of all, Propp is paying attention to 'things that happen', what here I will call 'events' (§2.1). This is the 'does what' of 'who does what to whom'. He is furthermore sensitive to the position of events in the timeline of the tale. Second, he is paying attention to the 'who' and 'whom' as well: the agents and patients (§2.2). Third, he is paying attention to the 'meaning' of the acts, both in the sense of the 'surface' meaning of the event (is the verb used 'give' or 'marry'?) and its meaning for the deeper purpose of the act in the context of the tale (§2.3).

Much of this information cannot be reliably extracted automatically and directly from the texts; that is, our automated natural language processing technologies are not up to the task of producing error-free interpretations for the above-mentioned layers of information. Therefore, this information was extracted in several different ways, usually relying on some amount of human attention and correction, as summarized later in Table 7. Layers marked 'Automatic' means the layer was calculated automatically by machine and not corrected or adjusted by hand. 'Automatic, with corrections' means that the layer was automatically calculated, and was corrected unilaterally by the annotation manager when an error was discovered. 'Semi-Automatic' means the layer was first annotated automatically, and then these annotations were hand-corrected by human annotators in a double-blind, adjudicated procedure. 'Manual' means the layer was annotated completely from scratch by hand in a double-blind, adjudicated procedure. The details of the annotation process are described in Section 4. Importantly, all these techniques for extracting this information relied on a significant amount of syntactic preprocessing (§2.4).

Finally, to measure the quality of the automatic learning algorithm, we needed an explicit representation of Propp's actual morphology—the gold standard answer, if you will (§2.5).

For each of the layers described below, there is an accompanying annotation guide (see §4), sometimes running to thirty pages, that describes the layer in great detail (these guides are included in the corpus release described in §5).

2.1 Events and the Timeline

To extract and represent the timeline of the story, I used an established representation suite, TimeML (Pustejovsky *et al.*, 2003; Saurí *et al.*, 2006). TimeML comprises three representations: events, time expressions, and time links. The first two mark the objects that populate the timeline, and the last defines the order of those objects on the timeline.

2.1.1 *Events*

Events are central to Propp's morphology. In TimeML, events are defined as happenings or states. They can be punctual, as in (1), or they can last for a period of time, as in (2). For the most part, circumstances in which something obtains or holds true, such as 'shortage' in (3) are considered events.

(1) Ivan struck the dragon's head from its body. (Punctual)

(2) The heroes traveled to far way lands. (Extended)

(3) There was a shortage of food across the kingdom. (Stative)

In addition to marking the presence of an event in the text by identifying the words that express the event, events are marked as one of seven different types: Occurrence, Reporting, Perception, Aspectual, Intensional Action, State, and Intensional State. These types are defined in the original TimeML annotation guide (Saurí *et al.*, 2006) and affect, among other things, how the event should be integrated into a representation of the timeline of the story.

2.1.2 *Temporal expressions*

In addition to events, TimeML provides for marking of temporal expressions, which indicate points or durations of time. Each expression is a sequence of words, potentially discontinuous, that indicate a time or date, how long something lasted, or how often something occurs. Temporal expressions may be calendar dates, times of day, or durations, such as periods of hours, days, or even centuries. Interestingly, time expressions are extremely sparse in these folktales, with only 142 instances over the whole corpus, averaging to only 7.5 time expressions per 1,000 words. Indeed, most of the tales had fewer than ten time expressions, and two had only a single one. This unexpected fact is perhaps due to folktales generally occurring on unspecified dates, or altogether outside of history. Regardless of the reason, time expressions proved to have little importance for the timelines as a whole.

2.1.3 *Temporal relationships*

Beyond what things happen (events) and points of time (temporal expressions), Propp was interested in the order of events in a tale. The TimeML standard provides for marking this information as well, in the form of 'time links'. A time link is a relationship between two times, two events, or an event and a time. It indicates that a particular temporal relationship holds between the two, for example, they happen one before another, as in (4).

(4) Ivan arrived before the Tzar. (Temporal: Before)

Time links fall into three major categories, each of which has a number of subtypes as outlined in the TimeML annotation guide (Saurí *et al.*, 2006). 'Temporal' links indicate a strict ordering between two times, two events, or a time and an event, as in (4). Six of the temporal links are inverses of other links (e.g. 'After' is the inverse of 'Before'; 'Includes' is the inverse of 'Included By', and so forth). Annotators used one side of the pair preferentially (e.g. 'Before' was preferred over 'After'), unless the specific type was specifically lexicalized in the text. 'Aspectual' links indicate a relationship between an event and one its subparts, as in (5). 'Subordinating' links indicate relationships involving events that take arguments, as in (6). Good examples of subordinating links are events that impose some truth-condition on their arguments, or imply that their arguments are about future or possible worlds.

(5) Ivan and the dragon began to fight. (Aspectual)

(6) Ivan's brothers forgot to wake him. (Subordinating)

2.2 Referential structure

In addition to events, Propp was sensitive to actors: the agents and patients of the events in the story. The raw information for representing actors is given by referring expression and co-reference chains (Hervás and Finlayson, 2010). The semantic role representation (Palmer *et al.*, 2005) is used to link co-reference chains to the semantic subjects and objects of events.

2.2.1 *Referring expressions*

The referring expression layer marks collections of words that 'refer' to something. Two referential expressions are underlined in (7). In this sentence, both referents are people—concrete things in the story world.

(7) The Tzar kissed the Tzarina.

This simple example covers a large number of cases, but anything that can be referred to is potentially a referring expression. Importantly referents may or may not have physical existence, as in (8), may not exist at all, as in (9), or may even be events or times, as in (10).

(8) Ivan had an idea.
(9) If Ivan had had a horse$_1$, it$_1$ would have been white.
(10) Ivan traveled$_1$ to a faraway Kingdom. It$_1$ took a long time.

Generally, if something is referred to using a noun phrase, it was marked as a referent. This definition has the convenient property of making us mark events (such as 'traveled' above) only when they are picked out further beyond their use as a verb.

2.2.2 *Co-reference chains*

Examples (9) and (10) also illustrate an important and obvious point, namely, that a single referent can be mentioned more than once in a text. In each case above, there is a single referent with two referring expressions. These two referring expressions are 'co-referential' because they refer to the same referent. To build referents, collections of referring expressions that all refer to the same thing are brought together into a co-reference chain. Therefore, a co-reference chain is a list of referring expressions referring to the same thing.

2.2.3 *Semantic roles*

Referents, as captured by referring expressions and co-reference relationships, would not be of much use to a Proppian analysis if we did not know in which events they participated. To connect referents to events, I used the well-known semantic role labeling scheme known as PropBank (Palmer *et al.*, 2005). This annotation was performed semi-automatically, the automatic portion by a basic statistical semantic role labeler modeled on the analyzers described elsewhere (Gildea and Jurafsky, 2002; Pradhan *et al.*, 2005). This labeler was run over the texts to create argument boundaries and semantic role labels for each verb. Each verb was assigned a PropBank 'frame', which is a list of up to five roles and their descriptions. A list of generalized functions of each of the five roles (plus two additional role types that apply to all verbs) is given in Table 1.

The identity of the frame was the only piece of information not automatically annotated by the labeler. Annotators were required to add the frame, as well as missing arguments and semantic role labels, and to correct the extant argument boundaries and labels. Sometimes, an appropriate frame was not available in the PropBank frame set, and in these cases, the annotators found the closest matching frame and assigned that instead.

The PropBank annotation scheme assigns a set of arguments (spans of words, potentially discontinuous) to each verb in the text, along with a primary category role for each argument. A simple example of such a marking is shown in Example (11), where the verb is underlined, and the arguments are marked with brackets; the role label is shown. This example also illustrates the use of an ARGA, which is used in the case when the ARG0 argument is not the agent of the action, as in verbs that take causative constructions.

(11) [The guard]$_{\text{ARGA (agent)}}$ marched [Ivan]$_{\text{ARG0 (marcher)}}$ to [the gallows]$_{\text{ARGM-LOC (location)}}$.

Table 1 Generalized meanings of PropBank frame arguments

Role label	Generalized meaning
ARG0	Subject, agent, or theme
ARG1	Object or patient
ARG2	Instrument
ARG3	Start state or starting point
ARG4	Benefactive, end state, or ending point
ARG5	Direction or attribute
ARGM	Modifying argument, usually augmented with a feature; all verbs may take ARGMs regardless of their frame
ARGA	Agentive argument where the agent is not ARG0; see example

In addition to a label, each argument can be marked with a second tag, called the 'feature'. Features mark an argument as fulfilling a common role for a verb, such as providing a direction (DIR), location (LOC), manner (MNR), negation (NEG), or modality (MOD), among others.

2.3 Semantics

Events, referents, and their relationships form the basic skeleton of the surface semantics of the text. It gives us the rough form of the 'who does what to whom'. Beyond this, though, we need to express the actual meaning of the words in the text, as well as other relationships between events and referents. To this end, formalized representations of meaning were captured in four additional layers. One, Wordnet senses, is an established way of marking word meaning in relationship to an extant ontology (Fellbaum, 1998). The remaining three layers were developed specifically for this work, and were intended to capture aspects of the surface semantics critical to automating Propp's analysis.

2.3.1 *Wordnet senses*

Word sense disambiguation (WSD) (Agirre and Edmonds, 2007) is a well-known natural language processing task. In it, each word is assigned a single sense from a sense inventory. For the ProppLearner corpus, I used Wordnet version 3.0, which is a useful sense inventory because it has a significant amount of semantic information both included in the inventory itself (in the form of meaning-to-meaning relationships), as well as linked from external databases.

Although some WSD algorithms can perform rather well, for a fine-grained sense inventory such as Wordnet 3.0, most algorithms are not much better than the default most-frequent-sense baseline. Because of this, and because word meaning is so critical to Propp's analysis, annotation of word sense was done completely manually. While Wordnet's coverage is excellent, it occasionally lacks an appropriate word sense. In those cases, the annotators found a reasonable synonym and substituted that sense. In the rare case that they could not find an appropriate substitute, annotators were allowed to mark 'no appropriate sense available'.

2.3.2 *Referent attributes*

An attribute is anything that describes a single referent irrespective of anything else (i.e. not involving a relationship). An attribute can be delivered in a copular construction, such as in (12), or as part of a compound noun phrase, as in (13). Importantly, attributes are defined to be 'permanent' properties of the referent in question, meaning that they should not change over the timeline of the story. If they did change, they were considered TimeML states, and were annotated as such.

(12) Ivan is brave.
(13) The sharp sword.

Each attribute was additionally assigned one of thirteen tags, which are listed in Table 2. The tag allowed the post-processing to augment the description of the referent appropriately.

2.3.3 *Context relationships*

This representation marked static relationships between referents in the text. Like the semantic role representation, a particular expression was marked as anchoring the context relationship, such as 'siblings' in (14). Referents participating in that relationship were marked with roles relative to the anchor. Role marking could be a Wordnet sense or a PropBank role, as the annotators saw fit. In (14), 'Jack' would be marked with the Wordnet sense for 'brother', and Jill with the sense for 'sister'. Implicit relationships (i.e. without an anchor) could be marked as well, as in (15), where the fact that 'They' is equivalent to the set {*Jack*, *Jill*} can be marked by tagging 'They' with the Wordnet sense for 'set', and both 'Jack' and 'Jill' with the sense for 'member'.

Table 2 Categories of referent attributes and their meaning

Type	Description
Physical	Visible or measurable characteristics such as size, height, and weight
Material	What a referent is made or composed of, or one of its ingredients
Location	Identifying spatial position of a referent, e.g. 'His front teeth'
Personality	Nonphysical character traits of characters
Name/Title	Nicknames, proper names, titles, and other terms of address
Class	Answers the question 'What kind?'
Origin	Whence an object comes, e.g. 'Cockroach milk'
Whole	What the referent is (or was formerly) a part of
Ordinal	Indicates the order or position of the referent in a set
Quantification	Answers the question 'Which one(s)?'
Mass amount	Answers the question 'How much?'
Countable amount	Specific numbers that answer the question 'How many?'
Descriptive	Catch-all for attributes that do not fall into another category

(14) Jack and Jill were siblings.
(15) Jack and Jill went up the hill. They fetched a pail of water.

Allowing annotators to mark relationship role-fillers with either Wordnet senses or PropBank roles allowed the representation to cover relationships not only where there was a specific instantiation of the role in the lexicon, but also relationship roles that were more easily expressed as a role to a verb. For example, in (16), 'chicken legs', which fill the role of 'thing stood on' might be termed the 'prop', but what about the hut? Is there a good noun meaning 'thing being held up'? Even if we can find an appropriate single sense in Wordnet to cover this particular role, there is no guarantee that we will be able to find one for every role in a relationship.

(16) . . . a little hut that stood on chicken legs. . .

Because this representation was developed specifically for this work, automatic analyzers were not available, and therefore this representation was annotated manually.

2.3.4 *Event valences*

Event valence indicates how positive or negative an event is for the hero. This is important piece of information for Propp’s morphology because it gives us information about the meaning of this event in the context of the story. It is akin to Wendy Lehnert’s positive or negative mental states (Lehnert, 1981). The valence scale ran from −3 to +3, including 0 (neutral) as a potential valence, rather than being restricted to just positive or negative as in Lehnert’s representation. The import of each valence on the scale is laid out in Table 3.

Valences were annotated to fill a specific gap in the representation suite. In the course of developing the ASM algorithm, I noted the importance of inferred event valences to discovering the correct functions. Many times, information necessary to discover the function was implicit (namely, not explicitly mentioned in the story text itself). Usually, this information was of a commonsense nature, such that the idea that ‘if someone is murdered, it is bad for them’, or that, given a choice, ‘people usually choose pleasure over pain’. Commonsense reasoning of this sort is a topic of active research, where even state-of-the-art inference engines and databases are not equal to the task of inferring the information needed even in this simple context. Therefore, instead of trying to extract this information automatically, I approached this as an annotation task, much like all the other information collected in the corpus.

2.4 Syntax

Because a number of the above representational layers were calculated automatically, or semi-automatically, I also added a number of syntax representations. Syntax representations are the scaffolding on which the semantic representations were built. For the purposes of the Propp study, these representations are not interesting *per se*, but rather are mainly useful for calculating the semantic representations. These layers included: Tokens, Part of Speech Tags, Multi-word Expressions (MWEs), Sentences, Lemmas, and Context-Free Grammar Parses.

2.4.1 *Tokens*

The first layer calculated for each text was the token representation. Tokens are defined as simple, unbroken spans of characters. A token marks the location of each word or word constituent, following the Penn Treebank tokenization conventions (Marcus *et al.*, 1993). Importantly, this convention marks contractions such as ‘n’t’ and “ve’ as their own tokens, but leaves hyphenated words as one large token. This representation was automatically calculated by the Stanford tokenizer (Manning *et al.*, 2014). Although the tokenizer is extremely accurate (greater than 99%), it still produced a few errors. As these were discovered, I corrected them myself across all versions of a text. Example (17) shows a tokenization of a sentence, where each of the eight tokens is underlined.

(17) He would n’t fight the three-headed dragon.

Table 3 Event valences and their meaning

Valence	Description	Example
−3	Immediately bad for the hero or his allies	The princess is kidnapped; the hero is banished
−2	May lead directly to a−3 event	The hero and the dragon fight
−1	Someone threatens a −2 or −3 event	The witch threatens death to, or chases, the hero
0	Neither good nor bad	
+1	Someone promises a +2 or +3 event	An old man promises help someday when most needed
+2	May lead directly to a +3 event	Someone hides the hero from pursuit
+3	Immediately good for the hero or his allies	The hero marries the princess; the hero is given gold

2.4.2 *Multi-word expressions*

MWEs are words that are made up of multiple tokens. MWEs are important because many appear independently in sense inventories like Wordnet, and they must be marked to attach word senses to them. Example (18) shows two types of continuous multi-words: a compound noun ('world record') and a proper noun ('Guinness Book of World Records').

(18) The world record is found in the Guinness Book of World Records.

MWEs may or may not have unrelated interstitial tokens. An example of a noncontinuous MWE, the verb-particle multi-word 'look up', is shown in (19).

(19) He looked the word up in the dictionary.

Although there are now detectors available for MWEs (Kulkarni and Finlayson, 2011), there were none available when the corpus was being constructed. Thus, the annotators were required to manually find and mark MWEs. This was not performed as a separate annotation task, but rather in the course of annotating word senses and semantic roles (see above).

2.4.3 *Part-of-speech tags*

Each token and MWE is tagged with a Penn Treebank part of speech tag (Marcus *et al.*, 1993). This representation was automatically calculated by the Stanford Part-of-Speech tagger (Manning *et al.*, 2014). The tagger has an accuracy greater than 98%. The annotator-corrected errors were corrected in the course of annotating word senses and semantic roles. Part-of-speech tags are fundamental for all other layers—they are important for identifying verbs for semantic role labeling, identifying nouns for use in referring expressions, identifying adjectives for attributes, and so forth.

2.4.4 *Lemmas*

Each token and MWE that is not already in root form is tagged with its lemma, or root form. This is a simple annotation which merely attaches a string to the token or MWE. This representation was automatically calculated using a Java implementation of the Wordnet stemmer morphy (Finlayson, 2014). The stemmer is reasonably accurate, and errors were corrected by the annotators in the course of annotating word senses and semantic roles.

2.4.5 *Sentences*

Sentences are important when calculating parse trees, which themselves can be used to calculate higher-level representations such as semantic roles. Sentences are merely lists of consecutive tokens, and they were automatically calculated by the Stanford CoreNLP sentence detector (Manning *et al.*, 2014). The sentence detector is extremely accurate, and the few errors were corrected manually.

2.5 Propp's morphology

To allow the end result of the morphology learning study to be evaluated, we needed Propp's own answers marked on the corpus, in addition to the raw data. For this purpose, I translated Propp's morphology into two representations, one for the 'functions' and another for the 'dramatis personae'.

2.5.1 *Dramatis personae*

Propp identified seven types of characters found in his folktales. This representation consisted of seven labels listed in Table 4. Any number of these could be attached to a particular referent in the text. Not all characters filled a 'dramatis personae' role, and in such cases, no tag was attached to that referent. In other cases, as Propp noted, a single character fulfilled more than one role.

Table 4 Propp's 'dramatis personae' and their meanings

Role	Description
Hero	main character of the story
Villain	perpetrator of the villainy; struggles with the Hero
Helper	accompanies and assists the Hero
Donor	prepares and provides the magical agent to the Hero
Princess	sought-for person, not necessarily female
Dispatcher	sends the Hero on his adventure
False Hero	someone who pretends to be the Hero to gain the promised reward

The label 'False Hero' did not occur in the corpus.

2.5.2 *Functions*

Annotating Propp's functions was a delicate task. While Propp described his morphology in great detail, it still was not specified in such a way as to allow unambiguous annotation in text. There are at least four main problems with Propp's scheme as described: unclear placement, implicit functions, inconsistent marking of trebling, and, in a small number of cases, apparent disagreement between Propp's own function descriptions and what is found in the tale.

With regards to unclear placement, consider, for example, the following excerpt of Afanas'ev's tale #148.

> *The tsar went in person to beg Nikita the Tanner to free his land from the wicked dragon and rescue the princess. At that moment Nikita was currying hides and held twelve hides in his hands; when he saw that the tsar in person had come to see him, he began to tremble with fear, his hands shook, and he tore the twelve hides. But no matter how much the tsar and tsarina entreated him, he refused to go forth against the dragon. So they gathered together five thousand little children and sent them to implore him, hoping that their tears would move him to pity. The little children came to Nikita and begged him with tears to go fight the dragon. Nikita himself began to shed tears when he saw theirs. He took twelve thousand pounds of hemp, tarred it with pitch, and wound it around himself so that the dragon could not devour him, then went forth to give him battle.*

Propp indicates the presence of functions B and C. Propp defines B as 'Misfortune or lack is made known; the hero is approached with a request or command; he is allowed to go or he is dispatched', with an abbreviated definition of 'mediation, the connective incident'. He defines C as 'The Seeker agrees to or decides upon counteraction', with an abbreviated definition of 'beginning counteraction'. Roughly, these two functions are the presentation of the task to the hero (B), and the acceptance of that task (C). Where exactly is B? Is it the whole section? Is it from the word entreated to the word begged? Should function boundaries correspond to sentence or paragraph boundaries? Are the children 'dramatis personae' in this tale (Dispatchers?), or are they are merely instruments of the tsar and tsarina? Is their imploring to be considered part of B?

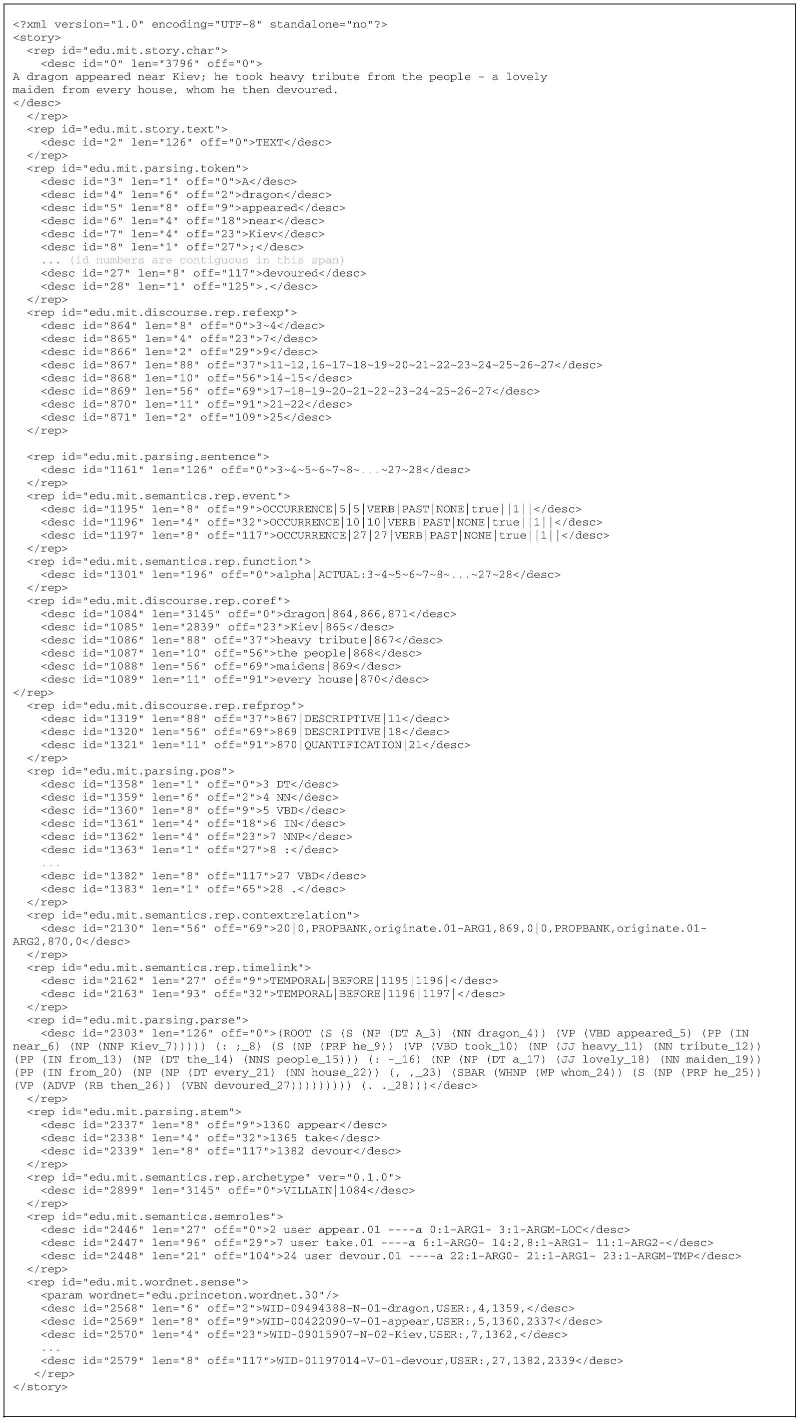
To address this problem, annotators marked two groups of tokens when identifying functions. First, they marked a region which captured the majority of the sense and extent of a function. This was usually a sentence, but extended to a paragraph or more in some cases. Second, they marked a defining word for the function, which usually took the form of single verb.

Implicit functions were the second problem; these were functions which were not lexicalized anywhere in the text. With regard the previous quote, one may ask 'Where exactly is C?' This is the decision to go forth against the dragon. It seems to happen somewhere between Nikita's shedding of tears and his preparation for battle by obtaining hemp, but it is not expressed anywhere directly in words; that is, the function is implicit. Propp notes that implicit functions occur frequently; yet, he gives no way to identify when they happen, and marks them inconsistently. To address this problem, when the annotators could find no set of words that captured a function that Propp indicated occurred in a tale (in his Function Table in his Appendix III), they chose the most closely logically related event and marked it with a tag, indicating it as an 'Antecedent' or a 'Subsequent', as appropriate.

With regards to inconsistently marked trebling (function groups that were repeated two, three, or four times in succession) or when indicated functions did not seem to match the tale itself, the annotators did their best to determine the correct marking. Fortunately, most of the time, typographical errors were restricted to disagreement in function subtypes, which does not directly impact the learning results based on the corpus.

2.6 Excerpt from the data

To give the reader a sense of the actual layout, format, and content of the annotation files, an excerpt of the first tale (Nikita the Tanner) is given in Fig. 1. This figure reproduces the xml markup associated with the first sentence of the



```
<?xml version="1.0" encoding="UTF-8" standalone="no"?>
<story>
  <rep id="edu.mit.story.char">
    <desc id="0" len="3796" off="0">
A dragon appeared near Kiev; he took heavy tribute from the people - a lovely
maiden from every house, whom he then devoured.
</desc>
  </rep>
  <rep id="edu.mit.story.text">
    <desc id="2" len="126" off="0">TEXT</desc>
  </rep>
  <rep id="edu.mit.parsing.token">
    <desc id="3" len="1" off="0">A</desc>
    <desc id="4" len="6" off="2">dragon</desc>
    <desc id="5" len="8" off="9">appeared</desc>
    <desc id="6" len="4" off="18">near</desc>
    <desc id="7" len="4" off="23">Kiev</desc>
    <desc id="8" len="1" off="27">;</desc>
    ... (id numbers are contiguous in this span)
    <desc id="27" len="8" off="117">devoured</desc>
    <desc id="28" len="1" off="125">.</desc>
  </rep>
  <rep id="edu.mit.discourse.rep.refexp">
    <desc id="864" len="8" off="0">3~4</desc>
    <desc id="865" len="4" off="23">7</desc>
    <desc id="866" len="2" off="29">9</desc>
    <desc id="867" len="88" off="37">11~12,16~17~18~19~20~21~22~23~24~25~26~27</desc>
    <desc id="868" len="10" off="56">14~15</desc>
    <desc id="869" len="56" off="69">17~18~19~20~21~22~23~24~25~26~27</desc>
    <desc id="870" len="11" off="91">21~22</desc>
    <desc id="871" len="2" off="109">25</desc>
  </rep>

  <rep id="edu.mit.parsing.sentence">
    <desc id="1161" len="126" off="0">3~4~5~6~7~8~...~27~28</desc>
  </rep>
  <rep id="edu.mit.semantics.rep.event">
    <desc id="1195" len="8" off="9">OCCURRENCE|5|5|VERB|PAST|NONE|true||1||</desc>
    <desc id="1196" len="4" off="32">OCCURRENCE|10|10|VERB|PAST|NONE|true||1||</desc>
    <desc id="1197" len="8" off="117">OCCURRENCE|27|27|VERB|PAST|NONE|true||1||</desc>
  </rep>
  <rep id="edu.mit.semantics.rep.function">
    <desc id="1301" len="196" off="0">alpha|ACTUAL:3~4~5~6~7~8~...~27~28</desc>
  </rep>
  <rep id="edu.mit.discourse.rep.coref">
    <desc id="1084" len="3145" off="0">dragon|864,866,871</desc>
    <desc id="1085" len="2839" off="23">Kiev|865</desc>
    <desc id="1086" len="88" off="37">heavy tribute|867</desc>
    <desc id="1087" len="10" off="56">the people|868</desc>
    <desc id="1088" len="56" off="69">maidens|869</desc>
    <desc id="1089" len="11" off="91">every house|870</desc>
</rep>
  <rep id="edu.mit.discourse.rep.refprop">
    <desc id="1319" len="88" off="37">867|DESCRIPTIVE|11</desc>
    <desc id="1320" len="56" off="69">869|DESCRIPTIVE|18</desc>
    <desc id="1321" len="11" off="91">870|QUANTIFICATION|21</desc>
  </rep>
  <rep id="edu.mit.parsing.pos">
    <desc id="1358" len="1" off="0">3 DT</desc>
    <desc id="1359" len="6" off="2">4 NN</desc>
    <desc id="1360" len="8" off="9">5 VBD</desc>
    <desc id="1361" len="4" off="18">6 IN</desc>
    <desc id="1362" len="4" off="23">7 NNP</desc>
    <desc id="1363" len="1" off="27">8 :</desc>
    ...
    <desc id="1382" len="8" off="117">27 VBD</desc>
    <desc id="1383" len="1" off="65">28 .</desc>
  </rep>
  <rep id="edu.mit.semantics.rep.contextrelation">
    <desc id="2130" len="56" off="69">20|0,PROPBANK,originate.01-ARG1,869,0|0,PROPBANK,originate.01-
ARG2,870,0</desc>
  </rep>
  <rep id="edu.mit.semantics.rep.timelink">
    <desc id="2162" len="27" off="9">TEMPORAL|BEFORE|1195|1196|</desc>
    <desc id="2163" len="93" off="32">TEMPORAL|BEFORE|1196|1197|</desc>
  </rep>
  <rep id="edu.mit.parsing.parse">
    <desc id="2303" len="126" off="0">(ROOT (S (S (NP (DT A_3) (NN dragon_4)) (VP (VBD appeared_5) (PP (IN
near_6) (NP (NNP Kiev_7))))) (: ;_8) (S (NP (PRP he_9)) (VP (VBD took_10) (NP (JJ heavy_11) (NN tribute_12))
(PP (IN from_13) (NP (DT the_14) (NNS people_15))) (: -_16) (NP (NP (DT a_17) (JJ lovely_18) (NN maiden_19))
(PP (IN from_20) (NP (NP (DT every_21) (NN house_22)) (, ,_23) (SBAR (WHNP (WP whom_24)) (S (NP (PRP he_25))
(VP (ADVP (RB then_26)) (VBN devoured_27))))))))) (. ._28)))</desc>
  </rep>
  <rep id="edu.mit.parsing.stem">
    <desc id="2337" len="8" off="9">1360 appear</desc>
    <desc id="2338" len="4" off="32">1365 take</desc>
    <desc id="2339" len="8" off="117">1382 devour</desc>
  </rep>
  <rep id="edu.mit.semantics.rep.archetype" ver="0.1.0">
    <desc id="2899" len="3145" off="0">VILLAIN|1084</desc>
  </rep>
  <rep id="edu.mit.semantics.semroles">
    <desc id="2446" len="27" off="0">2 user appear.01 ----a 0:1-ARG1- 3:1-ARGM-LOC</desc>
    <desc id="2447" len="96" off="29">7 user take.01 ----a 6:1-ARG0- 14:2,8:1-ARG1- 11:1-ARG2-</desc>
    <desc id="2448" len="21" off="104">24 user devour.01 ----a 22:1-ARG0- 21:1-ARG1- 23:1-ARGM-TMP</desc>
  </rep>
  <rep id="edu.mit.wordnet.sense">
    <param wordnet="edu.princeton.wordnet.30"/>
    <desc id="2568" len="6" off="2">WID-09494388-N-01-dragon,USER:,4,1359,</desc>
    <desc id="2569" len="8" off="9">WID-00422090-V-01-appear,USER:,5,1360,2337</desc>
    <desc id="2570" len="4" off="23">WID-09015907-N-02-Kiev,USER:,7,1362,</desc>
    ...
    <desc id="2579" len="8" off="117">WID-01197014-V-01-devour,USER:,27,1382,2339</desc>
   </rep>
</story>
```

Fig. 1 Edited excerpt of the ‘Nikita the Tanner’ data file. Red ellipses [. . .] indicate removal of data to improve readability. The collocation and valence layers are not shown, as they did not contain any annotations for the first sentence of the tale

Table 5 Tales in the corpus

Tale number	Russian title	English title	Number of words	Number of events
148	Никита кожемяка	Nikita the Tanner	646	104
113	Гуси-лебеди	The Magic Swan Geese	696	132
145	Семь симеонов	The Seven Simeons	725	121
163	Бухтан Бухтанович	Bukhtan Bukhtanovich	888	150
162	Хрустальная гора	The Crystal Mountain	989	150
151	Шабарша	Sharbarsha the Laborer	1,202	236
152	Иванко Медведко	Ivanko the Bear's Son	1,210	223
149	Змей и цыган	The Serpent and the Gypsy	1,210	250
135	Иван Попялов	Ivan Popyalov	1,228	220
131	Фролка-сидень	Frolka Stay-at-Home	1,388	248
108	Ивашко и ведьма	Ivashko and The Witch	1,448	276
154	Беглый солдат и черт	The Runaway Soldier and the Devil	1,698	317
114	Князь Данила-Говорила	Prince Danila Govorila	1,774	341
127	Купеческая дочь и служанка	The Merchant's Daughter and the Maidservant	1,794	331
140	Зорька, вечорка и полуночка	Dawn, Evening, and Midnight	1,934	339
Average			1,258	229
Sum			18,862	3,438

file. In brief, the xml file is organized as follows: there is a single top-level 'story' tag, with a number of 'rep' children tags, each of which corresponds to a layer of annotation. Each of those has a zero or more 'desc' tags which contain the data that represent an individual for that layer ('desc' is short for 'description', a.k.a., annotation). I removed several things from the file to improve readability: markup ('factory' and 'param' tags) that controls how the file is processed in the editor is not shown. Also left out are version attributes ('ver') that indicate which revision of a layer specification is being used. Repetitive entries, such as exhaustive annotations for tokens and parts of speech, were removed, as the reader can infer the pattern through careful inspection. Those interested in the details can refer to the file format specification included in the article's accompanying data set.

3 Selection of Texts

Propp analyzed a specific set of tales to derive his morphology. He took the first 100 folktales of a classic Russian folktale collection by Alexandr Afanas'ev (Afanas'ev, 1957). While Propp did his work in the original language of the tales, Russian, for practical reasons, I analyzed them in translation. Anthropologists have examined studying tales in translation, and the consensus is that, for structural analyses of the first order, the important semantic information of the tale comes across in the translation. 'If one translated a tale into another language, the tale structure and the essential features of the tale images would remaining the same [. . .]' (Fischer, 1963, p. 249).

Propp, in his Appendix III, provided function markings for about half of the tales he analyzed: in the English translation of Propp's work, there are only forty-five tales in the function table, with a small number of additional analyses distributed throughout the text. As explained elsewhere (Finlayson, 2015), I restricted myself to single-move tales, and so the set of possible candidates was further reduced; across several different translations of Propp, only twenty-one single-move tales with function analyses were provided. My ability to annotate this set was further reduced by both readily accessible high-quality translations and my annotation budget. In the end, I was left with fifteen single-move tales, listed in Table 5, for a total of 18,862 words.

4 Annotation Process

The annotation was conducted by twelve annotators split across eight teams. Each team consisted of two annotators and one adjudicator (some people worked on more than one team, or dropped out in the middle of the project), and each team was responsible for a different set of annotation layers, as shown in Table 6.

In cases where an established representation was being annotated, I prepared an annotation guide from the available material for the annotation team. In cases of representations developed anew for this work, I created the annotation guide from scratch.

Texts were split into batches of about 3,000 words and distributed to the teams on an as-needed basis, usually once every 1–3 weeks. The annotators would each annotate their assigned texts, producing two parallel sets of annotations. They would then meet with the adjudicator, sometimes in person, but more often via video conference. The adjudicator had typically worked previously as an annotator on the layers in question, and was somewhat more experienced in the process of annotation and details of the layer. The adjudicator then merged the annotator texts into an adjudication text, and this text was corrected by the adjudicator in consultation with the annotators during the adjudication meeting to produce the gold standard merged text. Annotation of the whole corpus took approximately 10 months, and involved over 3,000 man-hours of work.

Annotation was carried out entirely with the Story Workbench annotation tool (Finlayson, 2008, 2011). The Story Workbench is a platform for general text annotation. It is free, open-source, cross-platform, and user friendly. It provides support for annotating many different types of information (including all those mentioned in this article), as well as conducting annotation in a semi-automatic fashion, where initial annotations are generated by automatic analyzers and can be corrected by human annotators. Importantly, the workbench includes a number of tools that ease the annotation process. First, the user interface incorporates a fast feedback loop for giving annotators information on annotation validity: when an annotation is syntactically invalid, or semantically suspect, a warning or error is shown to the annotator, and they are prompted to correct it.

Table 6 Teams and the layers for which they were responsible

Team number	Layers
1	Word Senses, Part of Speech Tags, Lemmas, MWEs
2	Referring Expressions, Co-Reference Bundles
3	Time Expressions, Events
4	Semantic Roles
5	Temporal Links
6	Referent Properties, Context Relations
7	Event Valences
8	Dramatis Personae, Functions

The workbench also contains a tool for automatically merging annotations from different texts into one. This tool was used not only to produce the texts that were corrected during the adjudication meetings, but also to produce the final texts included in the corpus. The workbench is extensible at many different levels, admitting new annotation layers and automatic analyzers.

Because the annotation of some layers depended on other layers being complete, annotation was organized into a two-stage process. In this process, teams 1–4 would annotate and adjudicate a text, followed by teams 5–8. These texts were then merged together into the final gold standard texts that contained all layers of annotation, and whatever remaining inconsistencies were corrected by the annotation manager in consultation with the adjudicators.

4.1 Agreement measures

The quality of the annotations can be assessed by measuring inter-annotator agreement. The most uniform measure of agreement across the different representations is the F_1-measure, which is calculated in the standard way (Van Rijsbergen, 1979). I used the F_1-measure instead of the more common Kappa statistic (Carletta, 1996) because of the difficulty in calculating the chance-level of agreement for most of the representations. The F_1-measure is

Table 7 Representation layers applied to the corpus

Group	Number	Representation	Annotation style	Measure	Agreement
Syntax	1	Tokens	Automatic, w/corr.	–	–
	2	Part of Speech Tags	Semi-automatic	Strict F_1-measure	0.98
	3	Sentences	Automatic, w/corr.	–	–
	4	Lemmas	Semi-automatic	Strict F_1-measure	0.93
	5	Context-Free Grammar Parses	Automatic	–	–
Referential Structure	6	Referring Expressions	Manual	Strict F_1-measure	0.91
	7	Co-reference Bundles	Manual	Chance-adjusted Rand	0.85
Timeline	8	Time Expressions	Manual	Strict F_1-measure	0.59
	9	Events	Semi-automatic	Strict F_1-measure	0.69
	10	Temporal Relationships	Manual	Strict F_1-measure	0.66
Semantics	11	MWEs	Manual	Strict F_1-measure	0.68
	12	Wordnet Senses	Semi-automatic	Strict F_1-measure	0.78
	13	Semantic Roles	Semi-automatic	ARG [0–5] F_1-measure	0.60
	14*	Referent Attributes	Manual	Strict F_1-measure	0.72
	15*	Context Relationships	Manual	Strict F_1-measure	0.54
	16*	Event Valences	Semi-automatic	Strict F_1-measure	0.78
Propp	17*	Propp's Dramatis Personae	Manual	Strict F_1-measure	0.70
	18*	Propp's Functions	Manual	Region F_1-measure	0.71

Layers are arranged in five groups.

*Indicate representations that were developed specifically to support the extraction of Propp's Morphology.

a natural outgrowth of merging annotations done by both annotators, has a clear interpretation with regard to the data, and allows a more direct comparison between different representations. Table 7 summarizes the agreements for the different representations annotated either manually or semi-automatically.

There were three exceptions to the use of the F_1-measure. First, I used the chance-adjusted Rand index (Hubert and Arabie, 1985) for agreement between co-reference layers, which is a better measure that reflects partial agreement over long co-reference chains.

Second, a less-strict F_1-measure was used to assess the semantic role annotations. When performing this annotation, I was unaware that the original PropBank annotation project provided annotators with pre-marked argument boundary options (derived from the Penn Treebank corrected parse trees); this process allowed them to achieve a quite high inter-annotator agreement for argument boundaries. In contrast, our strict F_1-measure for semantic roles averaged only 0.36 across all texts, which is quite low. However, ignoring agreement between argument auxiliary features, the verb syntactic features, and arguments other than the core arguments reveals a higher agreement, indicating that the core goal of the representation—namely, capturing the agents and patients of actions—was successful.

Third, Propp's functions needed a special agreement measure that took into account the difficulty of translating Propp's monograph into a consistent annotation guide. Propp's monograph was not originally intended as a formal annotation at all, but rather a theory of narrative put forth in the 1920s, before there was even such a thing as computational linguistics (or, indeed, computers). Propp's theory is difficult to translate into a precise annotation specification. He was quite vague about the identity of many of his functions; his Appendix III has numerous inconsistencies and vagaries. These problems result in a strict F_1-measure agreement of only 0.22, which was quite low. This number did not seem to reflect the annotation team's intuition that agreement was actually fair to good, once minor variations were ignored. Instead of a strict measure, then, I formulated a more generous measure in which two function markings were considered to agree if there is a substantial (more than half) overlap in the function regions. This nets an agreement to 0.71, more in line with the team's observation that the annotators did actually agree in broad outlines.

In four cases, the agreement falls below 0.7. First, time expressions achieved a 0.66 F_1-measure. This is not especially worrisome because time expressions are quite sparse and were generally superfluous to the actual progression of the timeline. Second, context relationships achieved only an F_1-measure of 0.54. This is a bit troublesome, but in the actual algorithmic analysis, this representation was primarily used to substitute individuals for the groups of which they were a part; other types of relationships were not used. Moreover, the final timelines were inspected manually to make sure all participants in the final event representations were individuals, and when an agglomerate object was discovered, the annotations were corrected as appropriate. So, it is unlikely that this lesser agreement had a substantial effect on the accuracy of the results. Both time links and semantic roles achieved an F_1-measure of about 0.6. While these numbers are disappointingly low, these were also the two most complex representations in the suite. These lower agreement numbers naturally reflect the fact that the annotators had difficulty keeping all the complexities of these representations in mind.

5 Release of Data

This article is accompanied by an archive that contains the actual annotated files and supporting documentation. The archive may be downloaded from the MIT DSpace online library repository.[1] It contains several different types of files. First, it contains the annotation guides that were used to train the annotators. The guides are numbered to match the team numbers in Table 6. Included here are not only detailed guides for some layers, as produced by the original developers of the specification, but also our synopsis guides for each layer, which were used as a reference and further training material for the annotators. Also of interest are the general annotator and adjudicator training guides, which outline the general procedures followed by the teams when conducting annotation. Those who are organizing their own annotation projects may find this material useful.

Second, the archive contains a comprehensive manifest, in Excel spreadsheet format, listing the filenames, word counts, sources, types, and titles (in both Russian and English) of all the texts that are part of the corpus.

Finally, the archive contains the actual corpus data files, in Story Workbench format, an XML-encoded stand-off annotation scheme. The scheme is described in the file format specification file, also included in the archive. These files can be parsed with the aid of any normal XML reading software, or can be loaded and edited easily with the Story Workbench annotation tool, also freely available.

6 Contributions and Lessons Learned

I have described in detail the construction of a corpus of Russian folktales. The annotations applied to the corpus were intended to reflect the ‘surface semantics’ of the texts in such a way as to support the automatic extraction of Propp’s Morphology of the Folktale. The corpus contains 15 texts, 18,862 words, and 18 different layers of annotation. As I have shown elsewhere (Finlayson, 2013), this is the most deeply annotated narrative corpus ever created.

There were numerous lessons learned from this effort. The first is that clean, formal annotation of the sort of extensive surface semantics necessary to understand stories is no mean feat. The annotation of the corpus took approximately 10 months, involving twelve different trained annotators, adjudicators, and managers, and required over 3,000 man-hours of effort. The approximate cost of assembling these data was over $125,000 USD. This does not count the many years spent designing the tools for doing the annotation, and the many months spent choosing annotation layers, writing and editing the annotation guides, and training the annotators. Creating this corpus was expensive, time-consuming, and difficult.

The data collected were, without question, the key enablers for the task of automatically learning Propp’s morphology of the folktale. Without the data, and the rich formalization of the semantics of the tales, we would have had little chance of making any headway at all on the project. Nevertheless, the process pursued here, of careful, human-corrected annotation, is not scalable, and not feasible for most projects. Although this corpus is the most deeply annotated corpus of its kind yet assembled, it is still quite small by absolute measures. If we are to enable digital scholarship in the humanities that takes advantage of the type of text semantics used in this study, it is clear that we will need to invest in more confederated and automated solutions, working closely with fields such as machine learning, artificial intelligence, and computational linguistics, as well as perhaps crowdsourcing techniques, to produce workable data for these sorts of problems.

Regardless, this work, and the learning results that it enabled, showed that we have reached a point where sophisticated semantic analyses, previously the domain of humanist experts, can begin to be replicated to some degree by computer. This portends the dawn of a new era in the relationship of humanists to computing, where we move beyond mere word statistics to the inference of compelling and relevant semantics.

Note

[1] http://dspace.mit.edu/handle/1721.1/100054

Acknowledgments

The work described in this article was conducted at the Massachusetts Institute of Technology (MIT) Computer Science and Artificial Intelligence Laboratory (CSAIL) and funded by Defense Advances Research Projects Agency (DARPA) contract number FA8750-10-1-0076. During the preparation of this article, the author was partially supported by National Institutes of Health (NIH) grant number 5R01GM105033-02 and National Science Foundation (NSF) grant number CNS-1405863.

References

Afanas'ev, A. N. (1957). *Narodnye Russkie Skazki*. Moscow: Gos Izd-vo Khudozh Lit-ry.

Agirre, E. and Edmonds, P. (eds) (2007). *Word Sense Disambiguation Text, Speech, and Language Technology*. Dordrecht, The Netherlands: Springer.

Carletta, J. (1996). Assessing agreement on classification tasks: the Kappa statistic. *Computational Linguistics*, 22: 249–54. doi:10.1.1.48.4108.

Colby, B. N. (1973). A partial grammar of eskimo folktales. *American Anthropologist*, 75: 645–62.

Dundes, A. (1964). *The Morphology of North American Indian Folktales*. Vol. 195. Folklore Fellows Communications, Helsinki, Finland.

Fellbaum, C. (1998). *Wordnet: An Electronic Lexical Database*. Cambridge, MA: MIT Press.

Finlayson, M. A. (2008). Collecting semantics in the wild: the story workbench. In Beal, J., Bello, P., Cassimatis, N., Coen, M. and Winston, P. (eds), *Proceedings of the AAAI Fall Symposium on Naturally Inspired Artificial Intelligence (published as Technical Report FS-08-06, Papers from the AAAI Fall Symposium)*, Vol. 1. Arlington, VA; Menlo Park, CA: AAAI Press, pp. 46–53. http://www.aaai.org/Papers/Symposia/Fall/2008/FS-08-06/FS08-06-008.pdf

Finlayson, M. A. (2011). The story workbench: an extensible semi-automatic text annotation tool. In Tomai, E., Elson, D. and Rowe, J. (eds), *Proceedings of the 4th Workshop on Intelligent Narrative Technologies (INT4)*, Vol. 4. Stanford, CA; Menlo Park, CA: AAAI Press, pp. 21–4. http://aaai.org/ocs/index.php/AIIDE/AIIDE11WS/paper/view/4091/4455

Finlayson, M. A. (2013). A survey of corpora in computational and cognitive narrative science. *Sprache Und Datenverarbeitung (International Journal for Language Data Processing)*, 37(1–2). http://www.uvrr.de/index.php/anglistik/SDV_Vol_37_1-2_2013_Formal_and_Computational_Models_of_Narrative.html

Finlayson, M. A. (2014). Java libraries for accessing the Princeton wordnet: comparison and evaluation. In Orav, H., Fellbaum, C. and Vossen, P. (eds), *Proceedings of the 7th International Global WordNet Conference (GWC 2014)*. Tartu, Estonia: Global WordNet Association, pp. 78–85. http://gwc2014.ut.ee/proceedings_of_GWC_2014.pdf

Finlayson, M. A. (2016). Inferring Propp's functions from semantically annotated text. *Journal of American Folklore, Big Folklore: A Special Issue on Computational Folkloristics*, 192(511): 53–75.

Fischer, J. L. (1963). The sociopsychological analysis of folktales. *Current Anthropology*, 4: 235–95.

Gildea, D. and Jurafsky, D. (2002). Automatic labeling of semantic roles. *Computational Linguistics*, 28: 245–88. doi:10.1162/089120102760275983.

Hervás, R. and Finlayson, M. A. (2010). The prevalence of descriptive referring expressions in news and narrative. *Proceedings of the 48th Annual Meeting of the Association for Computational Linguistics (ACL 2010)*. Uppsala, Sweden: Association for Computational Linguistics (ACL), pp. 49–54. http://www.aclweb.org/anthology/P10-2010.pdf

Hubert, L. and Arabie, P. (1985). Comparing partitions. *Journal of Classification*, 2: 193–218.

Kulkarni, N. and Finlayson, M. A. (2011). jMWE: a Java toolkit for detecting multi-word expressions. In Kordoni, V., Ramisch, C..and Villavicencio, A. (eds), *Proceedings of the 8th Workshop on Multiword Expressions: From Parsing and Generation to the Real World (MWE 2011)*. Portland, OR: Association for Computational Linguistics (ACL), pp. 122–4. http://www.aclweb.org/anthology/W11-0818.pdf

Lehnert, W. G. (1981). Plot units and narrative summarization. *Cognitive Science*, 4: 293–331. doi:10.1207/s15516709cog0504.

Manning, C. D., Surdeanu, M., Bauer, J., Finkel, J., Bethard, S. J. and McClosky, D. (2014). The Stanford CoreNLP natural language processing toolkit. *Proceedings of 52nd Annual Meeting of the Association for Computational Linguistics (ACL 2014): System Demonstrations*, Baltimore, MD, pp. 55–60. http://www.aclweb.org/anthology/P/P14/P14-5010.pdf

Marcus, M. P., Marcinkiewicz, M. A. and Santorini, B. (1993). Building a large annotated corpus of English: the Penn treebank. *Computational Linguistics*, 19: 313–30.

Palmer, M., Kingsbury, P. and Gildea, D. (2005). The proposition bank: an annotated corpus of semantic roles. *Computational Linguistics*, 31: 71–105. doi:10.1162/0891201053630264.

Pradhan, S., Hacioglu, K., Krugler, V., Ward, W., Martin, J. H. and Jurafsky, D. (2005). Support vector learning for semantic argument classification. *Machine Learning*, 60: 11–39.

Propp, V. (1968). In Wager, L. A. (ed.), *Morphology of the Folktale*. 2nd edn. Austin, TX: University of Texas Press.

Pustejovsky, J., Castano, J., Ingria, R., Sauri, R., Gaizauskas, R., Setzer,A. and Katz, G. (2003). TimeML: robust specification of event and temporal expressions in text. In *Proceedings of the Fifth International Workshop on Computational Semantics (IWCS-5)*. Tilburg, The Netherlands.

Saurí, R., Littman, J., Knippen, B., Gaizauskas, R., Setzer, A. and Pustejovsky, J. (2006). *TimeML Annotation Guidelines, Version 1.2.1*. Waltham, MA. http://www.timeml.org/site/publications/timeMLdocs/annguide_1.2.1.pdf

Van Rijsbergen, C. J. (1979). Chapter 7: Evaluation. In *Information Retrieval*. London: Butterworths, pp. 112–40.